**Github**

Github adalah platform version control system, yang digunakan untuk menyimpan, sharing, berkolaborasi dengan tim dan memanajemen source code projek kita secara online kedalam sebuah repository. Analogi sederhananya seperti google drive. Bedanya, google drive biasa digunakan untuk menyimpan file-file, github dikhususkan untuk menyimpan kode-kode(source code) dari projek yang kita kerjakan. Selain itu, github memungkinkan kita melakukan sharing kode dari projek kita dan berkolaborasi dengan orang lain atau tim, jadi kita bisa bekerja bersama-sama di dalam suatu projek yang disimpan didalam repository github.

Cara menggunakan github antara lain:

1. Menginstall github <https://desktop.github.com/download/>
2. Konfigurasi github di laptop

Jika github sudah terinstall langkah selanjutnya adalah melakukan konfigurasi agar kita bisa mengupload projek kita ke github dengan perintah seperti ini:

* Buka cmd dengan tekan logo windows + R, ketik cmd lalu enter. Atau bisa menggunakan git bash dengan cara tekan logo windows dan cari git bash lalu buka aplikasinya
* Langkah selanjutnya adalah konfigurasi untuk menetapkan username dan email kita untuk informasi projek di github dengan perintah :  
  git config –global user.name username\_github

git config –global user.email email\_github

perintah diatas (tanpa tanda kurung) adalah menetapkan username dan email yang sudah kita daftarkan sebelumnya di github secara global, artinya --global adalah menetapkan akun author untuk semua projek yang kita kerjakan, jadi semua projek kita akan mendapatkan informasi tentang pembuat projek tersebut.

* Pergi ke github.com dan buatlah repository baru yang mana akan kita gunakan untuk menyimpan projek yang kita buat. Misalnya kita buat repo dengan nama “belajar-git”
* Setelah membuat/mengerjakan projek, langkah selanjutnya adalah melakukan inisiasi github di dalam projek kita supaya terkoneksi dengan github dengan cara:

git init

**git init** adalah perintah untuk menginstall github di projek kita, setelah melakukan init biasanya akan muncul folder baru dengan nama .git. folder ini berisi konfigurasi github di projek kita.

* Langkah selanjutnya adalah melakukan staging terhadap file-file didalam projek yang kita buat. Status **staging** artinya file-file tersebut sudah siap untuk di commit atau di upload dengan perintah:

git add nama\_file

atau git add .

**git add nama\_file** digunakan untuk menaikkan file yang sudah siap kita upload ke status staging JIKA hanya ingin mengupload file tertentu saja.

**git add .** digunakan untuk menaikkan semua file yang ada di projek kita ke status staging. Titik dalam perintah tersebut mewakili semua file yang ada di projek kita. Cara ini lebih cepat karena kita tidak harus satu-satu memanggil nama filenya.

* Setelah menaikkan semua perubahan atau file projek buatan kita ke staging. Langkah selanjutnya adalah melakukan commit dengan perintah:  
  git commit -m “pesan commit”

Commit adalah perintah untuk menyimpan projek yang kita buat ke dalam repository lokal(PC/laptop). Pesan commit biasanya diisi menyesuaikan dengan tugas yang kita lakukan misalkan kita mengedit projek maka contohnya “mengedit fitur A di halaman B”. -m adalah singkatan dari message untuk menyematkan pesan commit.

* Langkah selanjutnya adalah mengkoneksikan repo lokal ke repo online yang sudah kita buat sebelumnya yaitu belajar-git dengan perintah.

git remote add origin <https://github.com/Fatmarahma/belajar-git.git>

perintah adalah untuk mengkoneksikan repo lokal dan repo online, dimana projek yang kita buat akan tersimpan kedalam repo tersebut. Link yang tertera di akhir adalah link repo yang kita buat. Link tersebut akan otomatis kita dapatkan setelah membuat repository di github.

* Setelah langkah-langkah diatas dilakukan, langkah terakhir adalah mengupload projek yang tersimpan di repository lokal kita ke repository online yaitu github dengan cara:  
  git push origin nama\_branch

Perintah diatas adalah untuk mengupload projek kita ke repository online yang sudah kita buat sebelumnya. Nama branch adalah nama untuk bagian/cabang yang sudah kita kerjakan di projek tersebut.

**Branch** adalah konsep untuk mengelola perubahan yang terjadi di projek kita yang memungkinkan kita bekerja secara tim dengan masing-masing bagian/cabang. Hal ini berguna untuk memisahkan perubahan satu dengan lainnya tanpa mempengaruhi pekerjaan yang lainnya. Misal fatma dan jenni bekerja dalam satu projek, agar perubahan yang mereka lakukan tidak saling bertabrakan atau mempengaruhi satu sama lain, maka fatma dan jenni membuat branch masing-masing dengan tugas masing-masing. Dalam contoh ini fatma membuat branch dengan nama fatma/membuat-halaman-login lalu jenni membuat branch defngan nama jenni/membuat-halaman-register. Lalu mereka akan melakukan push dengan nama branch mereka masing-masing.

Contoh git push origin fatma/membuat-halaman-login .

dengan cara ini, pekerjaan yang dilakukan fatma dan jenni tidak saling mempengaruhi satu sama lain dan menghindari konflik kode seperti terhapusnya sebuah fitur atau bagian kodenya, dan lain sebagainya.

Untuk membuat branch sendiri, bisa menggunakan cara dibawah ini:  
cara pertama:

git branch nama-cabang

git checkout nama-cabang

cara kedua:

git checkout -b nama-cabang

kedua cara diatas sama tujuannya yaitu membuat cabang baru lalu mengarahkan dari cabang sebelumnya ke cabang baru yang sudah kita buat. Langkah membuat cabang ini dilakukan sebelum melakukan commit terhadap perubahan yang kita lakukan di projek kita.